



# ArcotID®

## Technical Whitepaper

*“Since the invention of public key cryptography twenty-five years ago, people have been struggling to secure the private key without the assistance of hardware. Arcot’s innovative Cryptographic Camouflage<sup>1</sup> has solved this problem. Finally there is a cost-effective and convenient means to strongly authenticate users and transactions over the Internet without the need for cumbersome hardware.”*

**Martin Hellman**  
Professor Emeritus,  
Stanford University

Organizations today have a wide variety of authentication methods available for their users. These range from the simple username/password mechanisms that exist in every operating system to one-time-password (OTP) tokens, biometric solutions and smartcard and software-based PKI systems.

However, virtually all of these solutions reflect the reality of an old security adage: “low-cost, easy and secure – choose two”. Security solutions tend to either be low-cost and easy to use and manage but insecure (such as username/password solutions) or very secure but expensive (such as OTP tokens and smartcards). Arcot now presents a third option: ArcotID, a software-based authentication system that provides both affordable security and ease of use.

### INTRODUCING ArcotID

The Arcot ArcotID is a “software smart card” that combines protection for Digital IDs approaching that of a hardware smart card with the lower cost and benefits of a software solution. Unlike traditional software key containers, ArcotID resists brute-force attacks using patented “Cryptographic Camouflage”<sup>1</sup> technology to hide the Digital ID from would-be attackers. The technology has been in use since 1997 and has been vetted academically by security experts and in practice by Fortune 500 companies.

ArcotID features an easy-to-use and familiar username-password or PIN-pad entry user interface and integrates quickly with existing infrastructures with support for standards such as RADIUS-based OTP, SAML, MS CSP and PKCS#11.

The ArcotID provides not only strong authentication but also enables PKI applications such as electronic document signing, secure email and secure e-commerce. As a software-based solution, ArcotID enables organizations to leverage the advantages of Public-Key Infrastructures without the expense and management issues inherent with hardware-based secure key storage.

### AN INTRODUCTION TO PUBLIC-KEY INFRASTRUCTURES

A Public Key Infrastructure (PKI) exists to provide secure online authentication services and bring the “print and sign” processes of the paper world to the digital world. Prior to public-key cryptography, authentication was generally based on the principle of a “shared secret”. This time-honored system of passwords, pass-phrases and secret handshakes required parties to arrange to share a piece of information that could mutually authenticate both sides of a transaction. The critical problem was (and continues) to be how to share a particular piece of information between parties when the number of participants can be unlimited. If every two parties need a shared secret between them, then the number of shared secrets grows at the rate of the square ( $N^2$ ) of the number of participants! This is obviously unworkable.

A better system is a central authority, trusted by all parties, that is responsible for authenticating every party and providing them with credentials that can be verified by anyone, based on the characteristics of the credential itself. A good example of this is a passport issued by the government. The government requires certain forms of proof of identity before issuing a passport and includes tamper-evident technology in the passport itself to reduce the probability of forgery. Once issued, the passport is a self-contained authentication credential.

### Public-Key Cryptography

The basis for PKIs is Public Key Cryptography, also known as “asymmetric key” cryptography. Public Key cryptography is a form of encryption where two mathematically related “keys” (seemingly random strings of numbers) can be used to encrypt (scramble) and decrypt (unscramble) messages and data using a computer. Messages encrypted with one key can only be decrypted with the other key and vice versa.

The crucial advantage of this property is realized when these keys are used in a very specific way. If one of the keys is kept secret by the owner but the other is tied to the owner’s identity, certified (notarized) by a trusted third party (similar to a government issuing a passport) and widely published, the infrastructure for digital signing is created. In the digital world, this is called a Public Key Infrastructure or PKI.

<sup>1</sup> See “Software Smart Cards via Cryptographic Camouflage”, D.N. Hoover and B. N. Kausik, Proceedings of the 1999 IEEE Symposium on Security and Privacy, IEEE Computer Society, Patent 6,170,058

In this scenario, if the secret or “private” key is used to encrypt a message, then only the widely published and certified key (or “public” key) will decode the message correctly. If one can be reasonably sure that the secret or private key was not stolen, then one can assume that the decrypted message was indeed sent by the person whose identity information is contained in the certified public key. This is the basis for *digital signatures*.

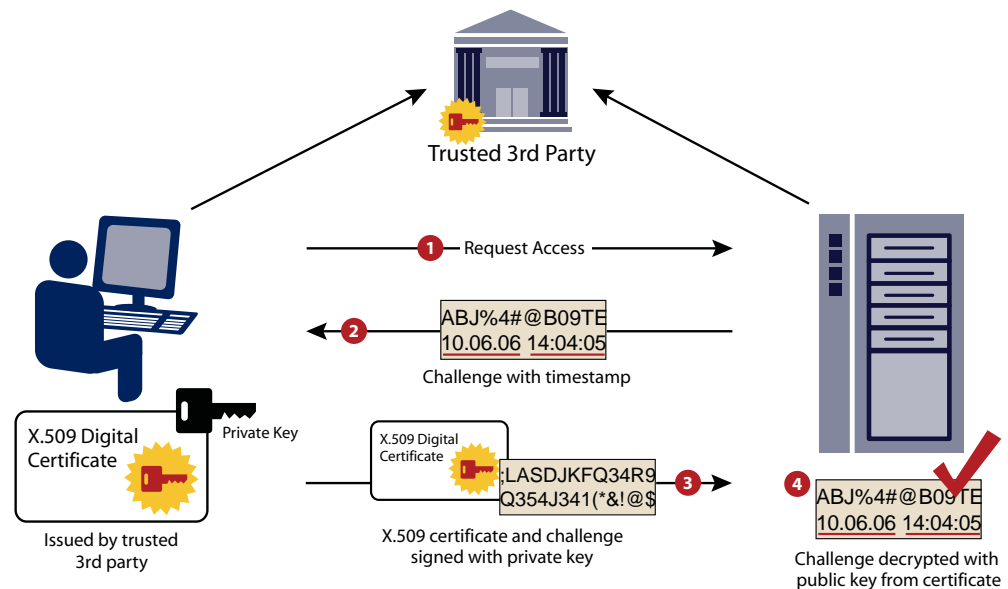
### Digital Signing and Authentication

Using additional algorithms such as *hashes* (see definition), PKI's can be used to digitally sign documents and provide the same or even stronger document integrity and authenticity guarantees than paper-based systems. Ultimately PKI's will enable businesses to replace “print and sign” processes with faster, more flexible and more efficient electronic equivalents.

Currently, the primary use of digital certificates is for authentication. The significant advantages of certificate-based authentication over other types of authentication are threefold: 1) it provides the strong security of two-factor authentication, 2) it is extremely scalable because it eliminates the need for a central directory and 3) it is resistant to man-in-the-middle attacks, most recently in the form of phishing attacks.

Technically, the advantages are that neither party needs to share any private information with the other party before or during the authentication process. This eliminates the risk of accidental disclosure of the password and also eliminates the  $N^2$  problem of shared secrets. PKI users can authenticate to other parties with whom they have no prior relationship by sending their credentials with the initial authentication request. This is similar to showing your ID with your ticket when boarding an aircraft.

A digital certificate-based authentication generally takes the form of a “challenge-response” transaction. A user requests a resource from a server. The server then responds with a “challenge” – a time-stamped random string of characters that the user receives and signs (encrypts) with their private key. The user sends the encrypted challenge back to the server along with their digital certificate. The server decrypts the challenge using the user’s public key contained in their certificate. If the decryption produces the original challenge, the user is authenticated and access is granted. This is shown below in Figure 1. In most secure transactions, the authentication is bi-directional; both sides will authenticate themselves using digital certificates.



**Figure 1** | Challenge-response authentication with Digital Certificates

### PROTECTING THE PRIVATE KEY

Regardless of the capabilities available to PKI users, PKI deployments still face a critical challenge – **securely and cost-effectively storing the private key**. In order to validate the authentication and maintain the assertion that a digital signature is binding on the signing party, there needs to be a high degree of confidence that the signing party is the only possible holder of the private key used for signing. If there is any doubt, then the possibility exists that an intruder can steal the key and masquerade as the legitimate key holder. Even if the key wasn't stolen, the signer can still claim that the key might have been stolen and the signature isn't his.

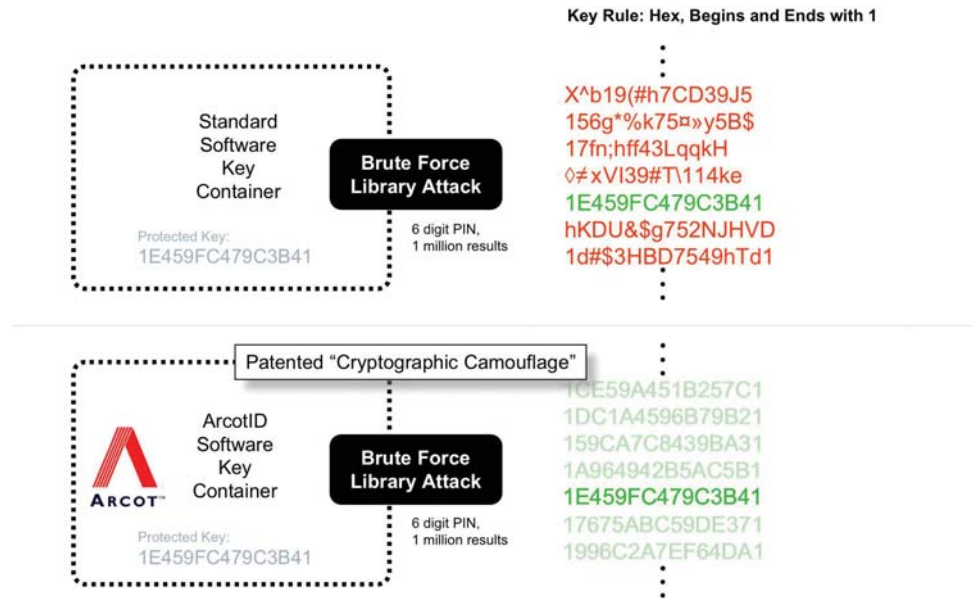
Keys can be stored in encrypted software modules, however they are subject to offline “brute-force” attacks that can be used to recover keys by exhaustively attempting all possible passwords. Current computers make this attack quite feasible, especially if users are using easy-to-remember words as passwords.

Secure storage is available in the form of Hardware Security Modules (HSM) and specialized smart cards called “crypto cards”, both of which permanently safeguard the private key in hardware. Unfortunately, both these solutions are expensive and require user training, extensive hardware deployments and replacement processes. These drawbacks have hindered general acceptance of PKI for digital signatures.

**ArcotID TECHNOLOGY**

The ArcotID was designed to solve this problem. An ArcotID, like a smart card or USB token, is a secure private key container, but delivered completely in software. However, unlike other software containers, the ArcotID is not subject to “brute force” attacks in which attackers copy the key container to their own equipment and exhaustively attempt millions of passwords which eventually leads to the disclosure of the private key.

The ArcotID prevents this and other attack scenarios with Arcot’s patented “Cryptographic Camouflage” technology. Using Arcot’s cryptographic camouflage technology, the key is encrypted based on the user’s PIN with standard encryption algorithms, but using the patented Arcot process. The effect of this process is that decryption, even using an incorrect PIN, will always produce a result that meets the specific, particular and well documented characteristics of a private key. So in the case of a simple 6-digit PIN, the brute force attack will produce approximately one million plausible, but invalid private keys. Keys produced as result of using an invalid PIN meet all the characteristics of a valid key, so they can be functionally used to encrypt or “sign” a challenge received from the Arcot server as a part of the authentication process. However, to prove that they have obtained the true private key protected within the ArcotID, the attacker must move further in the Arcot multi-key authentication process. The attacker is forced to encrypt/“sign” the challenge with the PIN decrypted key and respond to the Arcot authentication server which then determines whether or not a valid PIN was entered. If not, an invalid PIN counter is incremented and, just as with a hardware-based solution, the server can lock any access to the ArcotID container after a configurable number of invalid attempts. The attacker is able to attempt only a few passwords and is then prevented from further validation attempts. This thwarts offline brute-force attacks and creates the world’s first truly secure software key container.



**Figure 2 |** Effects of a brute force attack on a standard software key container and an ArcotID key container

**USING THE ArcotID FOR AUTHENTICATION**

Authentication using an ArcotID provides the benefit of a two-factor solution: authentication requires information only the user knows (a PIN or password) as well as the user’s ArcotID secure Digital ID container. As another layer of security, the user must also be able to contact their Arcot Authentication server in order to complete the authentication transaction.

The Arcot Authentication Suite supports a multitude of authentication schemes including client-side digital certificate-based authentication for web browsers and VPNs. Arcot supports standard integration APIs such as MS CSP and PKCS#11, one-time-password (OTP)-based authentication using the RADIUS protocol as well as native username/password authentication with Arcot libraries and APIs.

ArcotID Supported Authentication Models		
Authentication Type	Description	Sample Applications
Digital Certificate	PKI-based authentication using digital certificates. Applications natively support digital certificates accessed through standard interfaces (Microsoft CSP, PKCS#11)	MS Outlook, MS IE, Adobe Acrobat (Microsoft CSP), Netscape, Mozilla, Acrobat (PKCS#11)
Arcot Username/Password	Arcot native authentication method accessed with custom API’s or plug-ins. Uses challenge-response mechanism to authenticate user.	Custom VPN clients, internally developed applications using Arcot API, Arcot web server plug-ins.
One-Time-Password (OTP)	Arcot server authenticates user using native Arcot authentication and generates an OTP to be used for authentication to the application.	Applications that support RADIUS-based authentication. Includes examples such as VPNs, web applications, ERP applications and many others.

The Arcot Authentication server (WebFort) is described in detail in the Arcot WebFort White Paper.

**Using ArcotID natively (via API)**

The simplest form of authentication using the ArcotID is Arcot native authentication. In this case, the process starts when a user requests access to a protected asset, usually a web page but potentially any digital asset. The server redirects the request to the authentication plug-in which collects the userID from the user. The plug-in then sends an authentication request to the Authentication Server (WebFort). WebFort then creates a cryptographic challenge which is passed back to the software on the client machine along with the users Personal Assurance Message (or PAM, described in more detail later). This client software (a JavaScript applet on the login page or browser plug-in) presents the PAM and prompts the user for their password or PIN. The client software uses the PIN to decrypt the private key (which is cryptographically camouflaged in the ArcotID) and uses the now decrypted private key to encrypt the challenge sent by WebFort. The encrypted challenge is then sent to WebFort along with the user’s encrypted public key. The Authentication Server, which is the only server capable of decrypting the encrypted public key, uses the decrypted public key to verify the signed challenge and authenticate the user. If the authentication is successful, WebFort sends a message to the server plug-in to allow the user access. A simplified flow is shown in Figure 3 below.

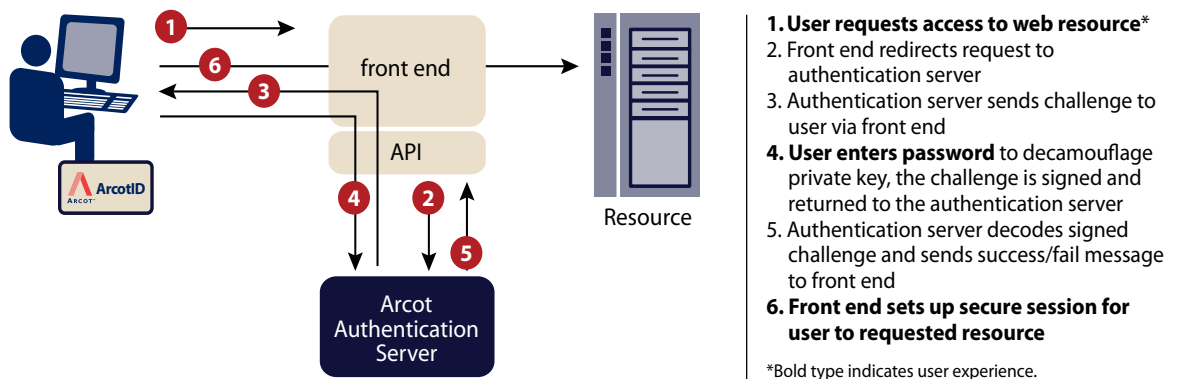


Figure 3 | Simplified Native ArcotID Authentication Flow

**Using ArcotID for OTP Generation with RADIUS**

Remote Authentication Dial-In User Service or RADIUS is an industry-standard interface for providing authentication, authorization and accounting (AAA) services to remote-access solutions. Using RADIUS, IT managers can deploy

remote access solutions and plug in their choice of authentication products. RADIUS is supported by virtually every remote access product currently available. The WebFort Authentication Server includes a RADIUS server which enables easy integration with existing access solutions.

RADIUS authentication is an example of “pass-through” authentication. The access device itself does not have native authentication capabilities so credentials provided by the user (for example, a username / password combination, username and OTP, signed challenge and certificate, O/S credentials token, etc.) are passed through to the authentication server for verification.

When a user with an ArcotID requests access to an asset protected by WebFort, the server protecting the asset has two options; one is to proxy a challenge/response transaction between the WebFort server and the client. The other is to redirect the authentication to the WebFort server. In the latter case, the WebFort server sends a cryptographic “challenge” directly back to the user’s machine. The Arcot client software, which can be as simple as a JavaScript web script, a VPN client plug-in, or a full-featured signing application prompts the user for his or her username and PIN. Using the private key decrypted by the PIN, the client software then signs the challenge sent by WebFort and returns it. If the PIN is correct and the challenge is properly signed, WebFort returns a One-Time-Password (OTP) to authenticate the user. In general, this OTP is automatically passed to the authorization server which passes the OTP back to WebFort via RADIUS. The user is approved by WebFort and the user authenticated message is returned to the authorization server. The server then permits the user to access the asset.

Despite the strong security and cryptographic complexity of the actual authentication, the user still sees a familiar username/password interface and the asset server integrates to a standard RADIUS server using OTP. This insistence on standard interfaces, both for end-users and IT processes is a critical factor in Arcot’s success.

This process is illustrated graphically in Figure 4. In this example, the user is attempting to authenticate to a VPN. The VPN Server is acting as the authorization server protecting the asset (the network). For more information on Arcot’s VPN authentication solutions, please see the Arcot VPN Authentication Solution Brief.

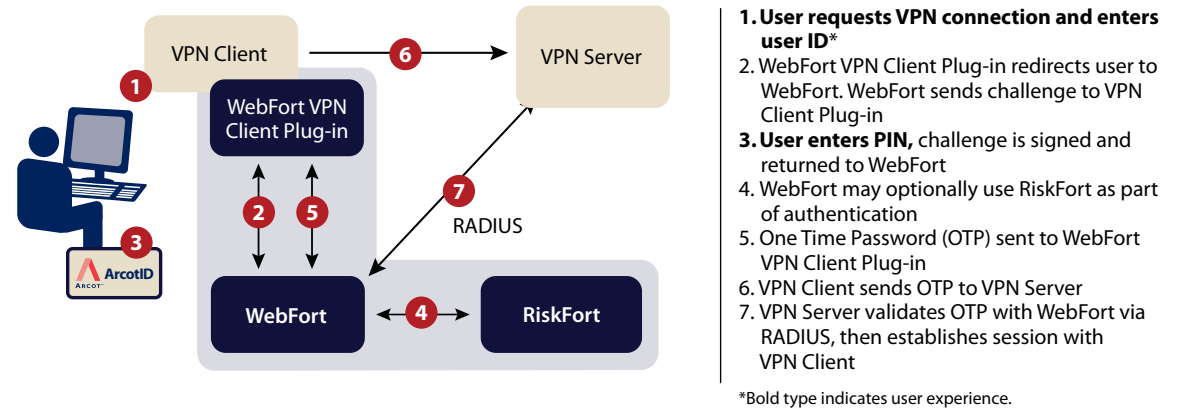
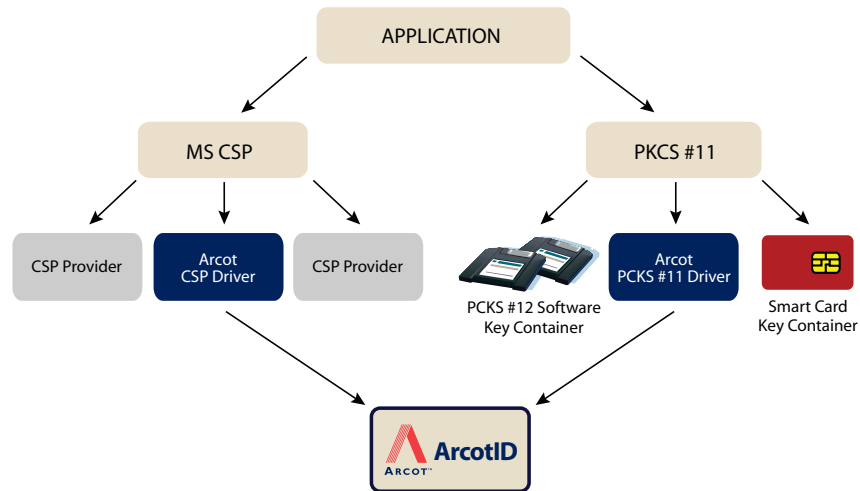


Figure 4 | VPN Authentication using Arcot OTP Authentication

### Using ArcotID to Authenticate with Digital IDs

Some applications such as web applications and VPNs have the ability to use client certificates or “Digital IDs” for user authentication. For these users, the problem of securely and cost-effectively protecting the private key is paramount. ArcotID functions as a strong software key container and can integrate seamlessly with these applications with both Microsoft Crypto Service Provider (MS CSP) and PKCS #11-compatible client software. Users can enjoy the same VPN or web interface they are accustomed to but gain the advantages of the ArcotID strong, two-factor security. Figure 5 shows how the ArcotID integrates with a PKI-enabled application. Note that using the solution requires no changes to the PKI-enabled application.



**Figure 5** | ArcotID Supports Industry Standard PKI APIs

A popular use of client certificates is authentication with “browser certificates” or certificates obtained with and stored in a web browser’s certificate store. For Internet Explorer, this certificate store is called a Crypto Service Provider (CSP) and is chosen when the certificate is initially issued. For Netscape-derived browsers such as Mozilla and Firefox, the default certificate store is a PKCS #12 certificate store. Both storage mediums are software-based and password-protected but unfortunately, both are subject to offline brute-force attacks.

When an ArcotID is used to store the credentials, the same certificates can be used in exactly the same manner using the same browsers but they will be protected using Arcot’s patented Cryptographic Camouflage, secure from attack. Details of the storage mechanism are provided in the Secure Digital ID Storage with ArcotID section below.

Implementing Digital ID-based authentication with ArcotID is similar to the native ArcotID authentication. Client machines must have the Arcot CSP or Arcot PKCS #11 driver installed. When the client application requests access to the private key contained in the ArcotID, the Arcot client software initiates a connection to the WebFort server requesting access to the key. The WebFort server issues a cryptographic challenge which is returned to the Arcot client software. The user is then prompted for their password or PIN to access the cryptographically camouflaged ArcotID private key. The challenge is encrypted with the de-camouflaged private key and sent back to the WebFort server. If the PIN or password is entered correctly, the challenge will be decrypted and the WebFort server will return a 3DES symmetric key used to decrypt the private key originally requested. The private key is then available for use by the CSP.

While the actual process to release the private key is very secure, the user experience remains a simple username/ password or PIN-pad interface.

### USING ArcotIDs FOR DIGITAL SIGNATURE APPLICATIONS

The truly unique feature of Public Key Infrastructures is enabling digital signing. The promise of moving physical “print and sign” processes to a pure digital document environment is an enormous driver for such paper-driven industries as pharmaceuticals, health-care, banking, lending and especially government. The secure storage of digital IDs is still a barrier to widespread adoption because of the potential for fraud and the subsequent loss of confidence due to compromised digital credentials. The ArcotID can be used to store third-party digital IDs (certificates and private keys) in a secure, confidence-building manner.

The Secure Digital ID Storage with ArcotID section below describes the mechanics of how a third-party certificate and private key can be stored in the ArcotID. Using a third-party certificate for digital signatures is however, a straight-forward process very similar to that described above in the authentication section. The primary difference is how the credentials are accessed.

There are two primary methods for accessing credentials for digital signatures. The first, as described above is implemented using an Arcot CSP or PKCS#11 driver. When the client application requests access to the private key contained in the ArcotID, the Arcot client software initiates a connection to the WebFort server requesting access to the key. The WebFort server issues a cryptographic challenge which is returned to the Arcot client software. The user is then prompted for their username and password or PIN to access the cryptographically camouflaged ArcotID

private key. The challenge is encrypted with the de-camouflaged private key and sent back to the WebFort server. If the PIN or password is entered correctly, the challenge will be decrypted and the WebFort server will return a 3DES symmetric key used to decrypt the private key originally requested. The private key is then available for use by the CSP and/or requesting application.

An example of this would be sending digitally signed email with Outlook. After composing the email, the user clicks the checkbox for a signed email and presses the send button. Instead of being prompted by the default CSP for a password to access the private key, the user is prompted for their ArcotID username and password or PIN. After the authentication process takes place with WebFort and the key is released, the Arcot CSP signs the message hash provided by Outlook and the signed email is sent. No changes are required to Outlook and the user interface remains familiar.

An example of the digital signature process is illustrated below in Figure 6 – Signing Adobe Acrobat documents using credentials stored in ArcotID. Note that only steps 1 and 3 involve user interaction – everything else happens automatically behind the scenes.

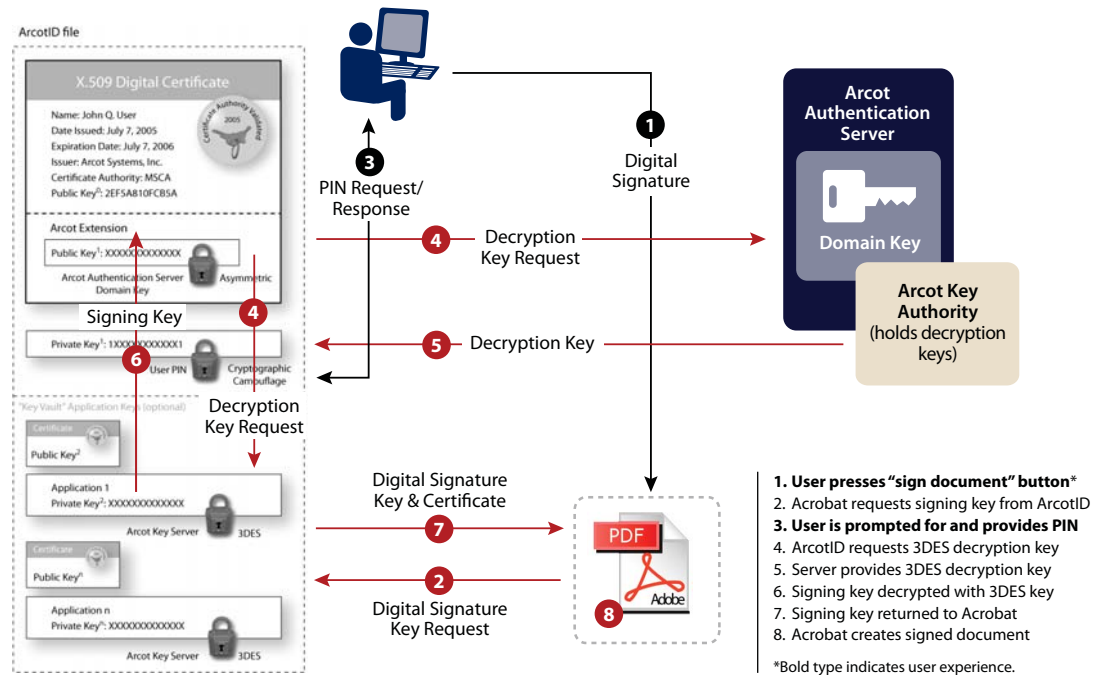


Figure 6 | Signing Adobe Acrobat documents using credentials stored in an ArcotID

### THE STRUCTURE OF AN ArcotID

An ArcotID is a software container formed of two components. The first component is always present and is created when the ArcotID is issued and assigned to an individual; the second is optional and exists only when an ArcotID is used to protect application specific credentials or Digital IDs, such as the private key used for digital signature applications.

The first component provides two core functions. The first function is to enable the strong authentication capabilities of the ArcotID. Secondly, it forms the basis for the secure software container or "Key Vault". This structure is shown in Figure 7.

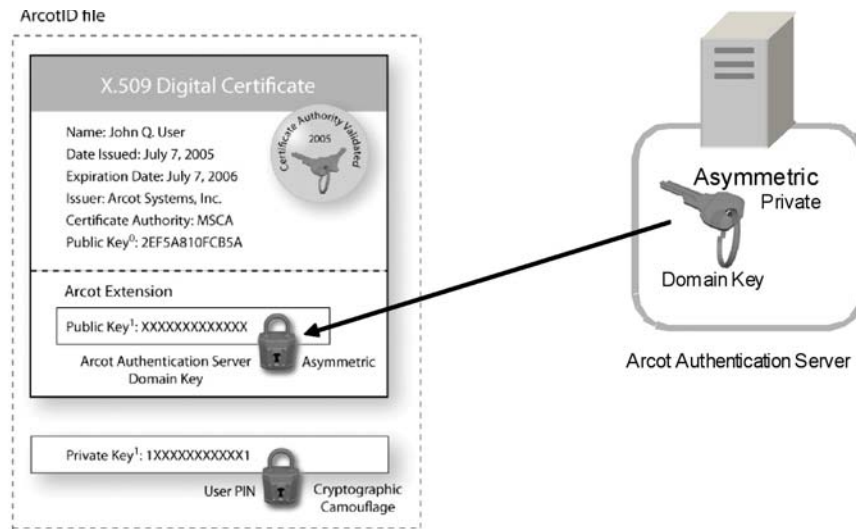
### CREATING AN ArcotID

The Arcot strong authentication solution utilizes asymmetric (public key) cryptography as the basis for its authentication method. To this end, a Certificate Authority is included as one of the components of the Arcot Authentication Solution, and is transparently used in the creation and validation of ArcotID credentials.

The first component or core of an ArcotID is formed as a standard X.509v3 digital certificate with an Arcot-specific extension. X.509 is a standard that defines the structure of digital certificates. According to the standard, an X.509 certificate contains identity information and a public key. Version 3 of the standard specifies how extensions may be added to the certificate structure. This is shown in Figure 7.

The certificate used in the ArcotID is created in such a way that only the public key used to form the basic X.509

certificate is retained. Because the certificate does not have a directly associated private key, its use can be limited to that of the domain where it was issued. This is an important feature of the Arcot Authentication Solution – unlike certificates issued by public CA’s, the issuer has full control over what purposes the key can be used for, and there is a strictly limited universe of valid users to notify in case any credentials are revoked. The fundamental purpose of this first ArcotID component is to enable strong authentication to the Arcot authentication server.



**Figure 7 | Structure of an ArcotID**

To complete the core certificate, a second key pair is generated which, like the first, is used exclusively for authentication purposes. This is shown in the figures as public key<sup>1</sup>. This public key is generated, encrypted with the domain key (a key known only to the Arcot Authentication Server), and then stored both on the certificate and the Arcot Authentication server. The private key is then “cryptographically camouflaged” using a derivative of the user’s PIN as the encryption key. Encrypting the public key with the domain key and “camouflaging” the private key with the user’s PIN implies that both the user’s PIN and a connection to the originating Arcot Authentication server are necessary to access both keys and complete an authentication transaction.

This completes the creation of a base ArcotID which is delivered to the user as a 2KB file. This ArcotID, in connection with the Arcot Authentication server, can now be used to provide strong user authentication to VPNs, Secure Portals, and other applications requiring user authentication.

**The Personal Assurance Message (PAM)**

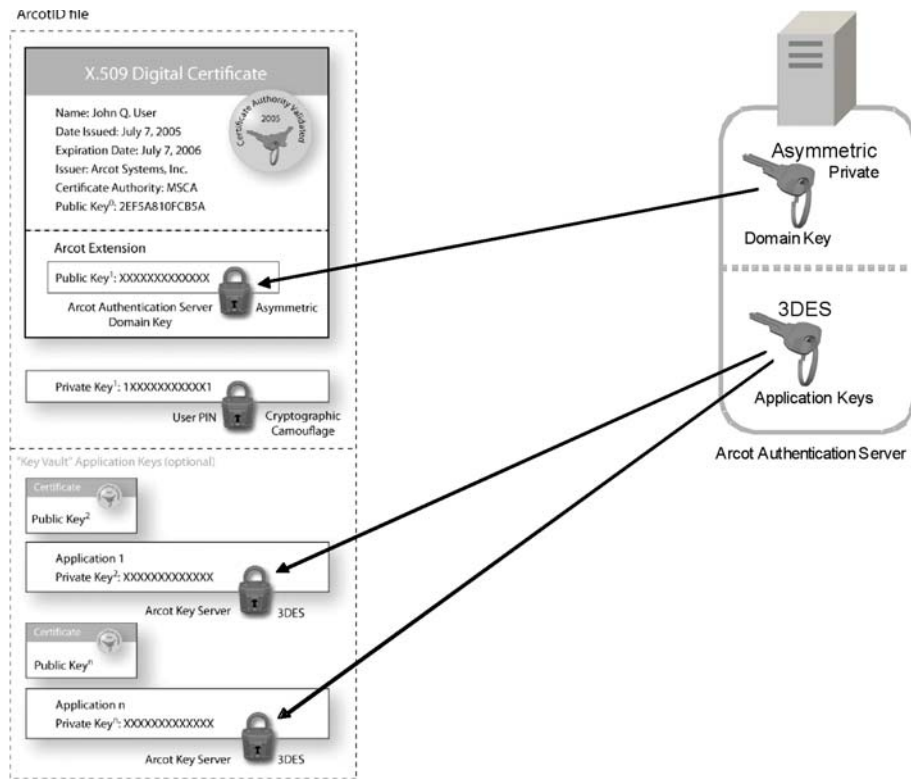
To further authenticate the server and thwart phishing attacks, users will choose a “personal assurance message” or PAM at enrollment time. This message is presented along with the password/PIN prompt to assure the user that they are dealing with an authorized web site. Since only the genuine Authentication Server has knowledge of this PAM, an imposter cannot successfully pretend to be the authentication server as in a phishing attack. Because of this feature, if a user does not see their PAM, they should suspect nefarious activity and abort their login attempt.

**SECURE DIGITAL ID STORAGE WITH ARCOTID**

**The ArcotID “Key Vault”**

After the core ArcotID is created, it can be used to store and protect application credentials or Digital IDs such as digital certificates and private keys. Digital certificates, by their nature, are stored but not encrypted. On the other hand, sensitive signing keys (private keys) are strongly encrypted in the ArcotID file.

Figure 8 illustrates the structure of the ArcotID with a “Key Vault”. In this case, a third-party Digital ID (certificate<sup>2</sup> and private key<sup>2</sup> in Figure 7) is stored in the ArcotID. The private key is encrypted by a key stored at the Arcot WebFort Key Authority server. The public key is also stored, but not encrypted in the key vault. This private key and certificate (or Digital ID) could be issued by a private PKI such as an IPSec VPN application or by a public CA such as Verisign or GTE.



**Figure 8** | Structure of an ArcotID with Application Certificates

The third-party Digital ID can be loaded onto an existing ArcotID or a new ArcotID generated for the Digital ID. Enrollment integration can be accomplished with the Arcot PKCS#11 driver or with Arcot’s API that integrates with popular PKI systems such as VeriSign OnSite, MS Certificate Server, Entrust, RSA Keon and CyberTrust Unicert.

When a Digital ID is loaded into the ArcotID, the Arcot Key Authority creates a unique symmetric key (3DES) to encrypt the application private key. The encrypted private key and plain-text certificate are stored in the ArcotID. The symmetric key used to encrypt the private key is stored in a high-security database at the Arcot Key Authority server.

As previously described in the section “Using ArcotIDs for Digital Signature Applications”, in order to access the third-party private key, the ArcotID will make a request for the symmetric key to the Authentication Server signed with the private key. The Authentication Server authenticates the request and passes it to the Key Authority. The Key Authority then securely sends the symmetric key back to the ArcotID holder. The ArcotID software uses the symmetric key to decrypt the application private key which is then available to applications for digital signing and other cryptographic operations. This complexity is completely transparent to the user who simply enters his or her PIN number in the familiar Arcot user interface to initiate the process. Everything else happens in the background without user intervention.

**CONCLUSION**

The Arcot ArcotID provides the most cost-effective, secure Digital ID storage available anywhere. Using patented *Cryptographic Camouflage*, the ArcotID provides unprecedented software security for digital credentials while preserving the ease-of-use critical to the success of any authentication solution. As part of the Arcot Authentication suite, ArcotID provides the strong basis for flexible authentication and digital signature solutions for any size enterprise or organization.

## GLOSSARY

### Digital ID

A Digital ID is the digital equivalent of a driver's license or passport. It contains information called a *Digital Certificate* which includes the holder's *identity* (name, email, address etc.) and is certified to be valid by a trusted third party (*Certificate Authority*). A Digital ID also contains a *Private Key* that can be used to *digitally sign* electronic documents.

### Private Key

A private key is used to *sign* an electronic document or any electronic data. The corresponding *Public Key* is used to verify the signature. Since anyone who possesses the private key can sign an electronic document with it, the private key must be very well protected to avoid others masquerading as the intended key holder.

### Public Key

A public key is used to verify a signature performed with a private key. In general, a public key is contained in a *Digital Certificate* along with *identity* information for the owner. In this way, a *digital signature* validated with a public key contained in a *digital certificate* can be used to verify who *signed* a digital document.

### Identity

The identifying characteristics of a user. These could include the user's name, phone number, email address, employer, country of residence or any other information appropriate to the application.

### Digital Certificate

A digital certificate contains a user's *public key* and *identity* information and is *signed* by a trusted third party known as a *Certificate Authority*.

### X.509v3 Digital Certificate

X.509 is a specification for digital certificates published by the ITU-T (International Telecommunications Union - Telecommunication). It specifies information and attributes required for the identification of a person or a computer system. Version 3 (X.509v3) defines the format for certificate extensions which can be used to add additional certified information to the certificate such as titles, authorities or other application specific data.

### Digital Signature

A digital signature is a combination of two processes. The first is creating a *hash* of the data or document to be signed. Then the user *signs* the hash using their *private key*. By validating the signature using the user's digital certificate and comparing the signed hash to a locally-calculated hash of the document or data, the receiving user verifies not just the identity of the signer but the integrity of the data or document.

### Signing

The act of encrypting a piece of data (usually a *hash*) with a *private key*.

### Encryption

Also known as symmetric encryption, it is the act of scrambling data so only the intended recipient can unscramble it. Both sender and recipient must share the same *encryption key* to scramble and unscramble the data.

### Public Key Encryption

Also known as asymmetric encryption, public key encryption is a form of encryption where there are two keys instead of one. One key, the *private key*, is used to encrypt (scramble) and the other, the *public key*, is used to decrypt (descramble).

### CA (Certificate Authority)

A certificate authority is responsible for *signing digital certificates*. Once a digital certificate is signed by a CA, anyone with the CA's digital certificate can verify the authenticity of any digital certificate signed by that CA. This process is also called "issuing digital certificates".

### PKI (Public Key Infrastructure)

A PKI is a phrase used to describe the entire infrastructure necessary to support the use of *digital certificates*. This includes a CA, directories for storing *identity* information and other services. PKIs are generally public (for example Verisign) or private (for example Entrust).

### Key Storage

Private keys are very sensitive to disclosure as their compromise can result in impersonation and fraud. Key storage ranges from simple software containers which are generally insecure to expensive but secure crypto-cards which perform all signing operations within the card. Arcot's ArcotID provides the security of a hardware smartcard with the ease of use of software.

### Revocation

When a user is no longer authorized to use a Digital ID, the ID must be revoked. This process can be difficult with standard PKIs however Arcot's system is designed to make this process easier.

### Non-repudiation

Non-repudiation is an important feature of PKIs. When used properly, A PKI-based system can prevent a user from denying responsibility for legitimate actions (signed a document, authenticated, sent a message etc.) that required them to use a digital ID. Since the user is the only one with access to the digital ID, they cannot disown actions taken with it.

### Hash

A hash, also known as a message digest, is a number generated from a string of text. The hash is substantially smaller than the text itself, and is generated by a formula in such a way that it is extremely unlikely that some other text will produce the same hash value. Hash functions are well-known and will always produce exactly the same result when performed on the same text.

### Cryptographic Camouflage

Arcot's patented cryptographic camouflage is used to protect Digital IDs stored in the ArcotID. Regardless of the PIN used to gain access, the result will always be a plausible key. This prevents users from attempting offline attacks on the ArcotID as they cannot know if they have discovered the right PIN without validating the result online. After a few failed attempts, the users account can be locked out, foiling any attempts to "crack" the ArcotID.

## ABOUT ARCOT SYSTEMS

Arcot is a leading provider of strong two-factor authentication and digital signature solutions. Arcot's software offerings are compliant with industry standards including 3-D Secure, IdenTrust, and SAFE. Over 11,000 banks, 25,000 merchants and other enterprises use Arcot to protect the identities of millions of employees, partners, and consumers worldwide. Organizations also work with Arcot to comply with regulations or legislation such as FFIEC, Sarbanes-Oxley, and HIPAA. Visit [www.arcot.com](http://www.arcot.com)

---

Corporate Headquarters | Arcot Systems, Inc. | 455 West Maude Avenue | Sunnyvale, CA 94085-3517 USA | [www.arcot.com](http://www.arcot.com) | (408) 969-6100  
 Arcot International | Trinity Court-A | Wokingham Road | Bracknell, Berkshire | RG42 1PL UK | +44 01344 668 280  
 Arcot Deutschland GmbH | Schulweg 7 | D-82343 Pöcking | Munich, Germany | +49 8157 997793

Arcot, the Arcot logo, ArcotID, Arcot KeyFort, Arcot RegFort, Arcot RiskFort, Arcot SignFort, Arcot TransFort, Arcot TrustFort, Arcot WebFort, Arcot Universal SAFE Signing Interface, and Arcot Universal Client, are trademarks or registered trademarks of Arcot Systems, Inc. in the United States and other countries. Other trademarks are the property of their respective owners.

Copyright © 2006 Arcot Systems, Inc. All rights reserved.

Part Number: WP100-10-0608

